

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Maja Zezula

## **Simulacija izmenjave sporočil v omrežjih VANET**

DIPLOMSKO DELO NA INTERDISCIPLINARNEM  
UNIVERZITETNEM ŠTUDIJU

MENTORICA: doc. dr. Mojca Ciglarič

Ljubljana, 2015



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Omrežja vozil (VANET) vedno bolj postajajo realnost. Specifika gibajočih se vozlišč in stalno spreminjajoče se topologije pa prinaša popolnoma nove komunikacijske in usmerjevalne probleme, ki jih še ne razumemo dobro. Preučite znanstveno ozadje problematike usmerjanja in topologije omrežij VANET. Izberite primerna orodja in okolja za simulacijo omrežij VANET in na preprostem primeru pošiljanja pozdravnih sporočil ter na realnem cestnem omrežju raziskujte lastnosti takšnega komunikacijskega omrežja. Opazujte predvsem zakasnitev, zanesljivost dostave sporočil in povezanost grafa vozil. Rezultate simulacij kritično ovrednotite.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Maja Zezula, z vpisno številko **63070002**, sem avtorica diplomskega dela z naslovom:

Simulacija izmenjave sporočil v omrežjih VANET

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Mojce Ciglarič,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 12. junij 2015

Podpis avtorice:





*Zahvaljujem se svoji mentorici, doc. dr. Mojci Ciglarič, za vse koristne nasvete, usmerjanje in pomoč pri izdelavi diplomske naloge. Zahvala gre tudi mojim staršem ter domačim za vso podporo tekom študija. Iskreno bi se zahvalila svojemu možu Urošu za podporo in vzpodbudne besede med časom študija in nastajanjem te diplomske naloge.*



# Kazalo

## Povzetek

## Abstract

<b>Poglavje 1</b>	<b>Uvod .....</b>	<b>1</b>
1.1	Motivacija .....	1
1.2	Omrežja vozil .....	2
1.3	Načini komunikacije v omrežjih VANET .....	2
<b>Poglavje 2</b>	<b>Izbira metod za simulacijo .....</b>	<b>5</b>
2.1	IEEE 1609 .....	5
2.2	Usmerjevalni protokoli .....	7
2.2.1	Topološko usmerjanje.....	8
2.2.2	Gručasto usmerjanje.....	11
2.2.3	Geografsko usmerjanje.....	11
2.2.4	Razpršeno usmerjanje.....	12
2.2.5	Pozicijsko usmerjanje .....	12
2.3	Generator prometa – SUMO .....	13
2.4	Izbira omrežnega simulatorja .....	14
2.4.1	NS-2 .....	14
2.4.2	OMNeT++ .....	15
<b>Poglavje 3</b>	<b>Priprava simulacijskega okolja .....</b>	<b>19</b>
3.1	OpenStreetMap.....	20
3.2	Konfiguracija generatorja prometa SUMO.....	21
3.3	Konfiguracija omrežnega simulatorja OMNeT++ .....	25
<b>Poglavje 4</b>	<b>Izvedba simulacije .....</b>	<b>29</b>
<b>Poglavje 5</b>	<b>Rezultati simulacij .....</b>	<b>31</b>
5.1	Prepustnost .....	32
5.2	Izgubljeni paketi .....	33
5.3	Zakasnitev od konca do konca .....	35
<b>Poglavje 6</b>	<b>Zaključek .....</b>	<b>37</b>



## Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>AODV</b>	Ad hoc On-Demand Vector	ad-hoc usmerjevalni protokol na zahtevo na osnovi vektorja razdalje
<b>AP</b>	Access Point	dostopovna točka
<b>DSRC</b>	Dedicated Short Range Communication	namenska povezava kratkega dosega
<b>MAC</b>	Media Access Control	dostop do medija
<b>MANET</b>	Mobile Ad-hoc Network	mobilno ad-hoc omrežje
<b>OBU</b>	On-Board Unit	sprejemna enota
<b>RSU</b>	Road Side Unit	obcestna enota
<b>TCP</b>	Transmission Control Protocol	protokol za nadzor prenosa
<b>TraCI</b>	Traffic Control Interface	kontrolni vmesnik prometa
<b>VANET</b>	Vehicular Ad-hoc Network	omrežje vozil
<b>WAVE</b>	Wireless Access in Vehicular Environment	brežžični dostop v avtomobilskem okolju
<b>XML</b>	Extensible Markup Language	razširljivi označevalni jezik



## **Povzetek**

Konstanten napredek v razvoju strojne, programske opreme in komunikacijskih tehnologij omogoča načrtovanje in implementacijo različnih tipov omrežij, ki jih je mogoče uporabiti v različnih okoljih. Eno izmed takšnih omrežij je tudi omrežje vozil (angl. Vehicular Ad-hoc Network, VANET). Cilj diplomskega dela je bilo ugotoviti, kolikšna mora biti intenzivnost pošiljanja pozdravnih sporočil pri določeni gostoti vozil, opremljenih s sprejemnimi enotami OBU v Celju, da dosežemo optimalno povezljivost vozil. Predstavljeni so načini komunikacije ter standardi in načini usmerjanja, ki se uporabljajo v omrežjih vozil. Simulacijsko okolje sestavlja simulator cestnega prometa SUMO in omrežni simulator OMNeT++, v simulacijah pa je bil uporabljen realni zemljevid mesta Celje. Priprava simulacijskega okolja in izvedba simulacije sta opisana v 3. oziroma 4. poglavju. Sestavljen je bil model omrežja vozil, ki nam omogoča vpogled v samo delovanje omrežja vozil, pripravo in izvedbo različnih testnih scenarijev ter različne možnosti analize rezultatov.

**Ključne besede:** simulacija, omrežja vozil, VANET, SUMO, OMNeT++.





## **Abstract**

Constant progress in development of hardware, software and communication technologies enables the design and implementation of different types of networks that can be used in various environments. One such network is Vehicular Ad-hoc Network (VANET). The objective of the thesis was the determination of the intensity of sending the beacon messages at a certain density of vehicles that are equipped with the OBU units to achieve optimal connectivity of vehicles in the city of Celje. The thesis presents methods of communication, standards and routing protocols which are used in Vehicular Networks. Simulation environment consists of SUMO road traffic simulator and OMNeT++ network simulator. A realistic map of the city of Celje was used in the simulations. Preparation of the simulation environment and the execution of the simulation are described in Chapters 3 and 4. A model of Vehicular Network was built, which allows us the insight of Vehicular Ad-hoc Network functioning, the preparation and execution of different test scenarios and various options of the results analysis.

**Keywords:** simulation, Vehicular Networks, VANET, SUMO, OMNeT++.



# Poglavje 1 Uvod

Združevanje računalniških, telekomunikacijskih omrežij in različnih storitev omogoča razvoj različnih vrst tehnologij VANET. Omrežje vozil je v zadnjih letih postalo aktivno področje raziskav, razvoja in standardizacije, saj ima ogromen potencial za izboljšanje varnosti in učinkovitosti prometa ter povečanje udobja vožnje tako za voznike kot tudi potnike. Raziskave temeljijo predvsem na oblikovanju in implementaciji omrežja vozil ter usmerjanju, oddajanju paketov, kvaliteti storitev in varnosti. Število registriranih vozil se vsako leto povečuje, kar lahko posledično privede tudi do povečanja števila prometnih nesreč. To število bi lahko zmanjšali s pomočjo brezžične tehnologije v vozilih, tako da bi si vozila med seboj avtomatsko pošiljala sporočila. Varnost je v omrežjih vozil izrednega pomena, zato morajo biti ta pred uporabo v realnem prometu temeljito preizkušena. Simulacije se tako v praksi s poskusi na realni opremi in vozilih izkažejo kot odlična izbira, saj jih je enostavneje in ceneje izvesti. Omogočajo nam preizkušanje različnih testnih scenarijev, novih tehnologij in optimizacijo delovanja omrežja še pred uporabo na cestah, saj lahko predvidimo obnašanje omrežja v različnih situacijah.

## 1.1 Motivacija

Ker se tehnologija VANET pospešeno razvija in bo gotovo kmalu tudi na voljo v Sloveniji, sem želela preizkusiti delovanje in obnašanje omrežja vozil pri njegovem uvajanju. Pri simulacijah sem uporabila zemljevid Celja in predvidela, da bo število vozil s potrebno opremo za komunikacijo med vozili počasi naraščalo.

Cilj diplomskega dela je bil raziskati, kako medsebojno vplivata gostota vozil, opremljenih z OBU-ji, in pogostost pošiljanja pozdravnih sporočil ter ugotoviti, kdaj je njihova kombinacija najbolj optimalna. Za nas najbolj optimalno rešitev predstavlja čim večja povezljivost vozil. To pomeni, da skušamo doseči maksimalno število sosednjih si vozil, skozi katere bi se izmenjale poslane informacije. Vozila, ki ne bi bila v dosegu, teh informacij ne bi prejela. Za doseg cilja sta bili potrebni priprava in izdelava čim bolj realne simulacije omrežja vozil.

Prvi del diplomskega dela opisuje metode za simulacijo. Predstavljeni so standardi in načini usmerjanja, ki se uporabljajo v omrežjih vozil. Predstavljena sta tudi gradnika našega simulacijskega okolja, simulator cestnega prometa SUMO in omrežni simulator OMNeT++. Osrednji del diplomskega dela opisuje model omrežja vozil, pripravo simulacijskega okolja in izvedbo simulacije. Sledi analiza dobljenih rezultatov, s katerimi smo ocenili zmogljivost simuliranega omrežja.

## 1.2 Omrežja vozil

Omrežja vozil VANET so ena izmed najhitreje rastočih tehnologij na področju komunikacijskih tehnologij. Sodijo v družino namenskih mobilnih ad-hoc omrežij (angl. *Mobile Ad-hoc Network*, *MANET*) le da so vozlišča vozila, ki se gibljejo z večjo hitrostjo kot vozlišča v omrežjih MANET. Ker so vozila omejena na ceste, je možno predvideti poti, po katerih se bodo gibala. Pri vožnji se lokacija vozil nenehno spreminja, prav tako pa se vozila gibljejo z različno dinamiko. To pomeni konstantno vzpostavljanje in rušenje povezav, povpraševanje po trenutni lokaciji in podatkih, kot so informacije o stanju na cesti, zastojih, možnih poteh, opozorila o zaviranju pred nami vozečih vozil, morebitnem trčenju, podrobna regionalna vremenska napoved ipd. V prometnih konicah se je tako moč izogniti zastojem in izbrati najbolj optimalno pot.

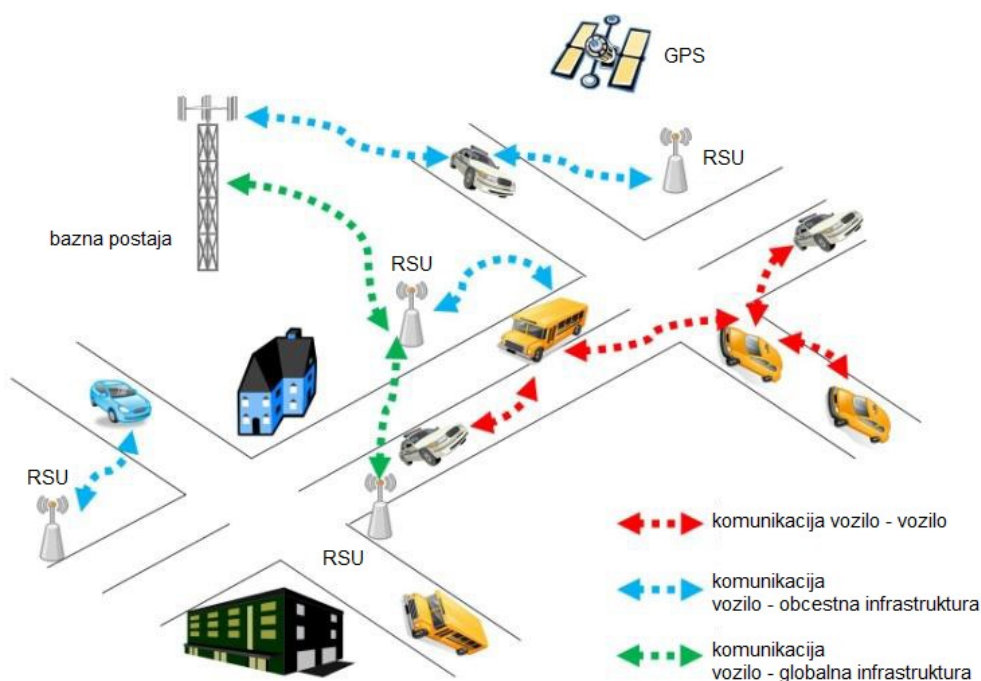
Vozila so opremljena s sprejemnimi enotami OBU (angl. *On-Board Unit*, *OBU*) kar jim omogoča medsebojno komunikacijo in izmenjavo podatkov. Ta funkcionalnost omogoča vrsto aplikacij za izboljšanje varnosti in učinkovitosti v prometu. Drugi ključni element v omrežjih VANET so infrastrukturne točke, t.i. obcestne enote RSU (angl. *Road Side Unit*, *RSU*), ki zagotavljajo povezavo z omrežnimi storitvami, podobno kot dostopovne točke (angl. *Access Point*, *AP*) v tradicionalnih brezžičnih omrežjih.

## 1.3 Načini komunikacije v omrežjih VANET

V omrežjih VANET lahko vozila komunicirajo neposredno med seboj ali z obcestno ter globalno infrastrukturo. Formacija takšnega omrežja, ki ga prikazuje Slika 1, omogoča nemoteno izmenjavo informacij, vsako vozilo, opremljeno s sprejemno napravo, pa je vozlišče v omrežju in lahko sprejema ter posreduje sporočila [3]. Način komunikacije se izbira glede na številne dejavnike, kot so npr. radijska

vidljivost, razpoložljivost komunikacijske infrastrukture, zahteve po kvaliteti storitev in nujnost komunikacije. V splošnem ločimo tri osnovne topologije povezovanja [4]:

- komunikacija vozilo – vozilo (angl. *Vehicle-to-Vehicle*, V2V),
- komunikacija vozilo – obcestna infrastruktura (angl. *Vehicle-to-Roadside*, V2R),
- komunikacija vozilo – globalna infrastruktura (angl. *Vehicle-to-Infrastructure*, V2I).



Slika 1.1: Formacija omrežja VANET [3].

Jedro omrežij VANET predstavlja komunikacija vozilo – vozilo, kjer vozila komunicirajo med seboj, komunikacija pa deluje v njihovi neposredni bližini. Večina kritičnih sporočil je vezana na radijski doseg komunikacije, vozila pa prevzemajo vloge komunikacijskih usmerjevalnikov ali končnih točk do tipične razdalje nekaj sto metrov [4].



## Poglavje 2 Izbira metod za simulacijo

Simulacije prometa nam omogočajo oceno infrastrukturnih sprememb še pred njihovo implementacijo v realnem svetu. Tako lahko, na primer, preizkusimo in optimiziramo učinkovitost okoljskih con ter obnašanje semaforiziranih križišč. S pomočjo simulacij lahko opazujemo obnašanje omrežja, simuliramo določene parametre in različne scenarije.

V spodnjih poglavjih so predstavljeni standardi in usmerjevalni protokoli, ki določajo način izmenjave informacij med vozili. Sledita opis simulatorja cestnega prometa SUMO in omrežnega simulatorja OMNeT++ ter opis ogrodij, ki smo jih uporabili pri simulaciji v omrežnem simulatorju.

### 2.1 IEEE 1609

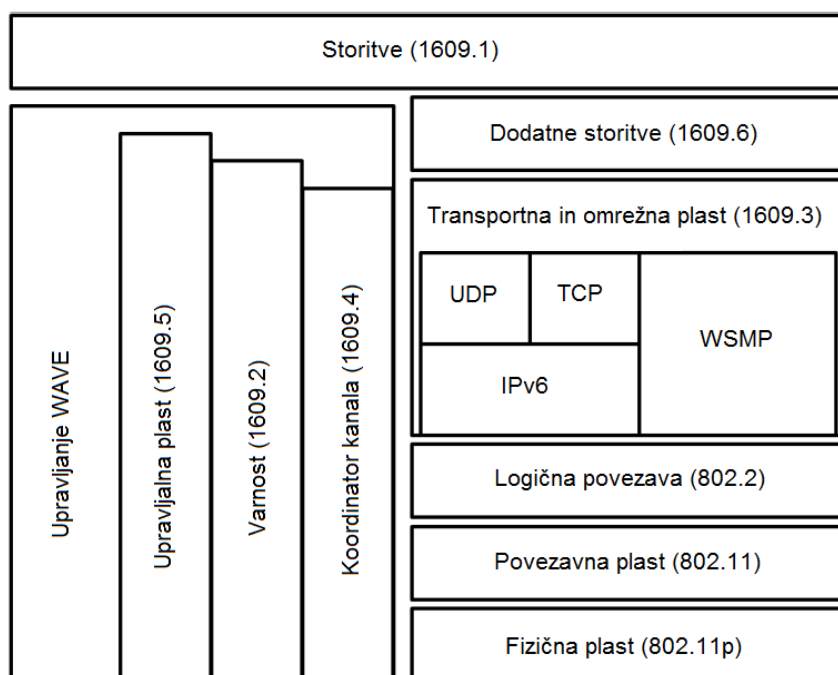
Standardi so dokumentirani dogovori, ki vsebujejo zahteve, tehnične specifikacije in druge kriterije, ki se uporabljajo kot pravila ali smernice, da bi izdelki, storitve in drugi procesi ustrezali svojim namenom. Pri postavljanju komunikacijskih sistemov je potrebno dosledno upoštevati in spoštovati področne standarde na vseh nivojih protokolnega sklada.

V primeru omrežja vozil, kjer morajo biti vozila mobilna preko prostranih območij, je združljivost v primerih gostovanja osnovni pogoj za zagotavljanje interoperabilnosti. Rešitev se vsaj na fizičnem sloju in dostopu do medija (angl. *Media Access Control, MAC*) kaže v različici poznane družine brezžičnih lokalnih omrežij (angl. *Wireless Local Area Network, WLAN*) iz delovne skupine IEEE 802.11p [4]. Kot nadgradnja standarda 802.11p je bil predlagan celovit sklad protokolne družine IEEE 1609 z imenom WAVE (angl. *Wireless Access in Vehicular Environment*). Protokolni sklad je prikazan na sliki 2.1.

Nabor standardov IEEE 1609 opisuje arhitekturo, komunikacijski model, protokole, varnostne mehanizme, omrežne storitve ter delovanje na fizičnem sloju in dostopu do medija v avtomobilskem okolju za doseganje velikih hitrosti (do 27 Mb/s) v brezžičnih komunikacijah kratkega dosega (do 1000 m). Standardi prav tako definirajo komponente sprejemnih in obcestnih enot. Razdeljeni so na šest področij:

- *IEEE 1609.1*: Opredeljuje storitve in njihovo delovanje ter interakcijo z obcestnimi enotami.
- *IEEE 1609.2*: Opredeljuje varnostne koncepte in obliko ter enkripcijo sporočil, pogoje za varno izmenjavo sporočil in njihovo obdelavo po prejemu.
- *IEEE 1609.3*: Opredeljuje transportno in omrežno plast vključno z naslavljanjem in usmerjevanjem. S protokolom WSMP (angl. *WAVE Short Messages Protocol*) skrbi za področje zagotavljanja varnosti.
- *IEEE 1609.4*: Opisuje izboljšave standarda IEEE 802.11 (dostop do medija) in opredeljuje večkanalno delovanje v brezžičnem dostopu v avtomobilskem okolju.
- *IEEE 1609.5*: Opisuje upravljalne storitve za brezžično povezovanje med vozili ter vozili in obcestnimi enotami.
- *IEEE 1609.6*: Opisuje dodatne storitve, ki omogočajo oddaljeno upravljanje podprtih naprav. Za področje varnosti uporablja storitve in protokol WSMP, ki je opredeljen v IEEE 1609.3.





Slika 2.1: Protokolni sklad WAVE.

Tehnologija WAVE je uvrščena v namenske povezave kratkega dosega (angl. *Dedicated Short Range Communication, DSRC*). Za hitro in zanesljivo povezavo ima dodeljen 75 MHz spekter v frekvenčnem pasu 5,9 GHz. Tehnologija DSRC omogoča razvoj storitev, kot so aktivna varnost v prometu, elektronsko cestninjenje, navigacija ipd., saj so standardi zasnovani tako, da izpolnjujejo zahteve storitev, ki so odvisne od izmenjave informacij med vozili ter vozili in obcestnimi enotami.

## 2.2 Usmerjevalni protokoli

V omrežjih vozil imajo usmerjevalni protokoli pomembno vlogo, saj določajo način izmenjave informacij med vozili. Usmerjevalni protokoli morajo biti sposobni dinamičnega ustvarjanja poti, jih med komunikacijo znati vzdrževati in v primeru napak ponovno vzpostaviti. Določiti morajo optimalne poti za zmanjšanje časovnih zamud pri usmerjanju. Ključni izziv pri oblikovanju usmerjevalnih protokolov je bil premagati te težave in zagotoviti komunikacijo z minimalnimi zakasnitvami in čim manjšo režijo.

V omrežjih vozil razdelimo usmerjevalne protokole v pet glavnih kategorij:

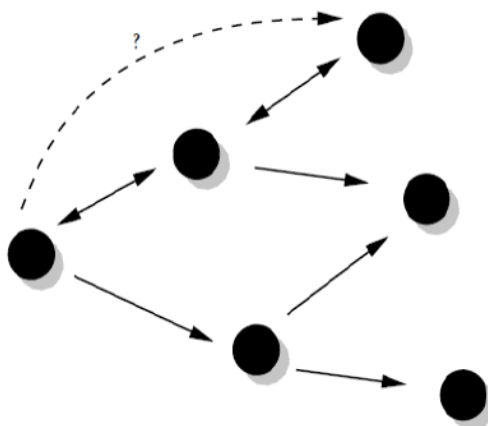
- *topološko,*
- *gručasto,*
- *geografsko,*
- *razpršeno in*
- *pozicijsko usmerjanje.*

### 2.2.1 Topološko usmerjanje

Usmerjanje paketov poteka na podlagi informacij komunikacijskih povezav med vozili. Topološko usmerjanje se deli na tri podskupine: reaktivno, proaktivno in hibridno usmerjanje.

#### 2.2.1.1 Reaktivno usmerjanje

Posredovanje paketov se prične ob zahtevi po komunikaciji z določenim vozilom. V omrežju se poišče pot s pošiljanjem poizvedovalnih paketov (angl. *Route Request Message, RREQ*), usmerjanje pa se zaključi, ko se vzpostavi pot do končnega vozila. Slabost protokolov je v preplavljanju paketov (angl. *flooding*) za odkrivanje poti, kar močno obremenjuje omrežje, saj se poveča režija in s tem pasovna širina.



Slika 2.2: Reaktivno usmerjanje [11].

Predstavnik reaktivnih usmerjevalnih protokolov je na primer ad-hoc usmerjevalni protokol na zahtevo na osnovi vektorja razdalje (angl. *Ad hoc On-Demand Vector, AODV*).

### AODV

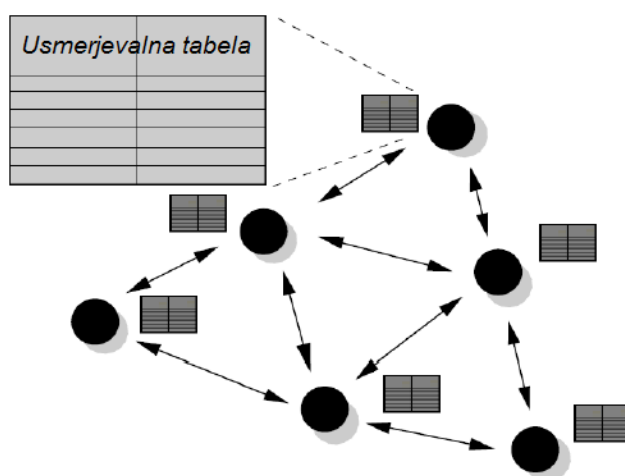
Protokol AODV, podobno kot ostali reaktivni usmerjevalni protokoli, vzpostavlja poti do končnih vozil le na zahtevo. Protokol omogoča dinamično, večskokovno usmerjevanje med sodelujočimi vozili, ki želijo vzpostaviti in vzdrževati ad hoc omrežje ter komunikacijo med dvema voziloma preko vmesnih vozil, če na primer ti dve vozili nista na istem območju. Ob poslanem zahtevku za povezavo se začne postopek odkrivanja najkrajše možne poti. Protokol AODV omogoča hiter odziv na spremembe poti. Ob prekinitvi povezave obvesti končno vozilo, neuporabljeno pot pa ovrže. Pot se šteje za aktivno, dokler se podatki med izvirnim in končnim vozilom izmenjujejo.

Glavna značilnost protokola AODV je uporaba zaporednih številok za vsak vnos poti pri naslovu končnih vozil. V primeru, da obstajata dve poti do končnega vozila, bo izvirno vozilo izbralo pot z večjo zaporedno številko. Protokol AODV za odkrivanje in vzdrževanje poti uporablja različna kontrolna sporočila:

- *RREQ*: Vozilo razpršeno oddaja pakete RREQ sosednjim vozilom. Ta paket potuje preko vmesnih vozil, dokler ne doseže končnega vozila. Paket RREQ vsebuje identifikator RREQ, ciljni IP naslov, ciljno zaporedno številko (angl. *Destination Sequence Number*), izvirni IP naslov in izvirno zaporedno številko (angl. *Originator Sequence Number*).
- *RREP*: Ko vmesno vozilo sprejme paket RREQ, pošlje paket RREP (angl. *Route Reply, RREP*) izvirnemu vozilu, da se preveri pot do končnega vozila oziroma posreduje paket RREQ končnemu vozilu, če ima podatke o novi poti. Paket RREP vsebuje informacije o skokih, ciljno zaporedno številko ter ciljni in izvirni IP naslov.
- *RERR*: Ob prekinitvi povezave se pošlje sporočilo o napaki (angl. *Route Error Message, RERR*). Če želi izvirno vozilo še vedno pošiljati podatke po tej poti, se ponovno začne postopek odkrivanja poti.

### 2.2.1.2 Proaktivno usmerjanje

Slika 2.3 prikazuje proaktivno usmerjanje. Vozila, ki uporabljajo proaktivne usmerjevalne protokole, nenehno posodablja usmerjevalno tabelo z informacijami o novih poteh znotraj omrežja s pošiljanjem pozdravnih sporočil ne glede na zahteve po komunikaciji. Podatki se tako lahko hitro posredujejo, ker iskanje novih poti ni potrebno, kar je zelo pomembno pri aplikacijah, ki zahtevajo minimalne zakasnitve. Slabost je seveda vzdrževanje tabele z neuporabljenimi potmi, kar močno poveča porabo pasovne širine.



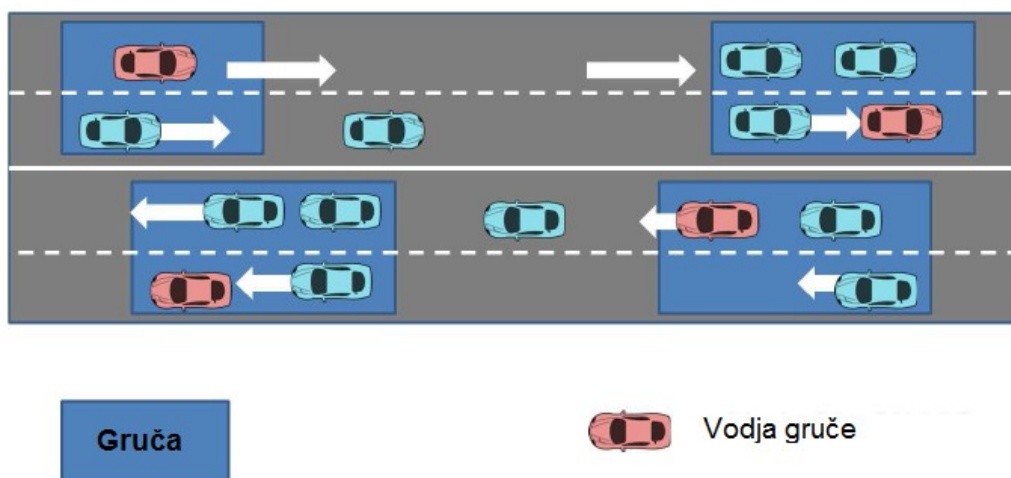
Slika 2.3: Proaktivno usmerjanje [11].

### 2.2.1.3 Hibridno usmerjanje

Hibridni usmerjevalni protokoli združujejo prednosti reaktivnih in proaktivnih protokolov. Omrežje se razdeli na dve plasti. Notranja plast je proaktivna, ki ohranja in periodično posodablja usmerjevalno tabelo med vozlišči glede na obremenitev omrežja, velikost pasovne širine in velikost omrežja. Zunanja plast je reaktivna, pot se vzpostavi na zahtevo. Tako se skrajša potreben čas za vzpostavitev poti.

### 2.2.2 Gručasto usmerjanje

Vozila, ki vozijo v bližini, v isti smeri in uporabljajo gručasto usmerjevalne protokole, ustvarijo skupine oziroma gruč (angl. *cluster*). V gruči se nato izbere vodjo (angl. *cluster head*). To vozilo je odgovorno za usmerjanje znotraj gruč in komunikacijo med sosednjimi gručami. Ker mora samo vodja gruč poiskati ciljno vozlišče, je režija pri usmerjanju manjša, vendar pa lahko pride do večjih zakasnitev ob formiranju gruč.



Slika 2.4: Gručasto usmerjanje [12].

### 2.2.3 Geografsko usmerjanje

Pri geografskem usmerjanju poteka usmerjanje od izvirnega vozila do vseh vozil znotraj določene geografske regije, imenovane interesna cona (angl. *Zone of Relevance, ZOR*). Preplavljanje paketov poteka samo znotraj interesne cone, s čimer se zmanjšata režija in obremenitev omrežja. Največja slabost geografskih usmerjevalnih protokolov je računanje dodatnih območij pokritosti oziroma delov omrežja, ki pa niso nujno povezani z ostalim delom omrežja.

### **2.2.4 Razpršeno usmerjanje**

Razpršeno usmerjanje je najbolj pogosto uporabljeno usmerjanje v omrežjih vozil. Paketi se s pomočjo preplavljanja pošiljajo po celotnem omrežju, s čimer se zagotovi dostava sporočil. Takšno usmerjanje je primerno za manjše število vozil v omrežju. Z večanjem števila vozil se namreč eksponentno poveča število sporočil, s tem pa tudi potrebna pasovna širina.

### **2.2.5 Pozicijsko usmerjanje**

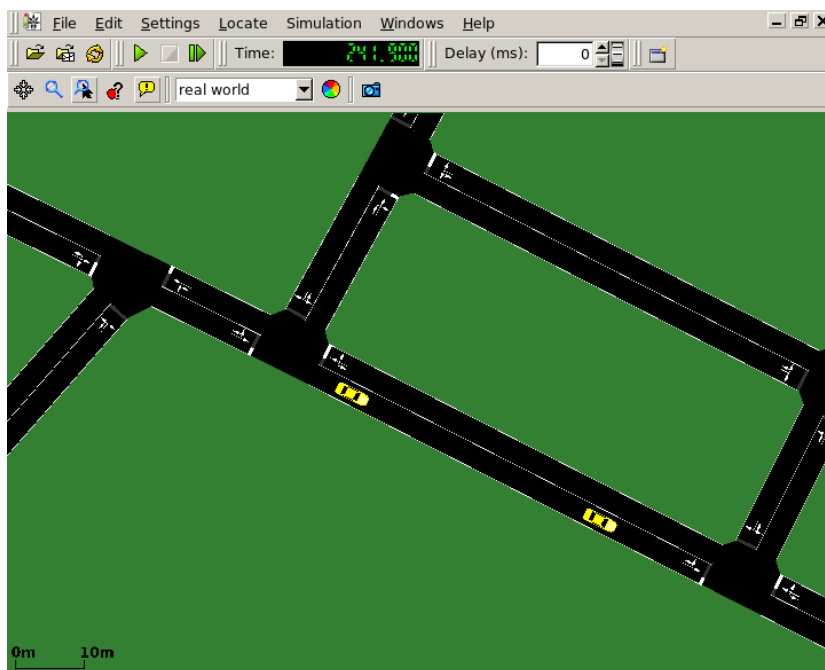
Pri pozicijskem usmerjanju se sporočila prenašajo na podlagi geografskega področja vozil. Podatki o lokaciji vozil se pridobijo iz globalnega pozicionirnega sistema (angl. *Global Positioning System, GPS*). Vsako vozilo mora poznati svojo lokacijo, ki jo doda v glavo sporočila in se uporabi pri usmerjanju paketov do končnega vozila. Pozicijsko usmerjevalni protokoli veljajo za robustne protokole, saj niti ne posodablajo usmerjevalne tabele niti si ne izmenjujejo informacij o stanju povezav.

## 2.3 Generator prometa – SUMO

Simulator SUMO (angl. *Simulation of Urban MObility, SUMO*) je odprtokodni, zelo zmogljiv in razširjen simulator cestnega prometa. Omogoča simulacije različnih prometnih sistemov, vključno s križišči, vozili, javnim prometom in pešci, delovanje semaforjev ipd. Vključena ima številna orodja, ki omogočajo vizualizacijo, uvoz omrežja in računanje emisij. Uporabljen je bil v številnih raziskovalnih projektih. V paketu SUMO lahko za izgradnjo cestnega omrežja uporabimo vgrajene aplikacije:

- *SUMO*: simulacija v ukazni vrstici,
- *GUISIM*: simulacija v grafičnem uporabniškem vmesniku,
- *NETCONVERT*: generator omrežja; izračuna in določi potrebne povezave,
- *NETGEN*: generator abstraktnih omrežij,
- *DUAROUTER*: generator poti v omrežju.

Za zagon simulacije v simulatorju SUMO potrebujemo vhodne datoteke, ki jih pripravimo s pomočjo razširljivega označevalnega jezika XML (angl. *Extensible Markup Language, XML*). Datoteke so predstavljene v poglavju 3.2.



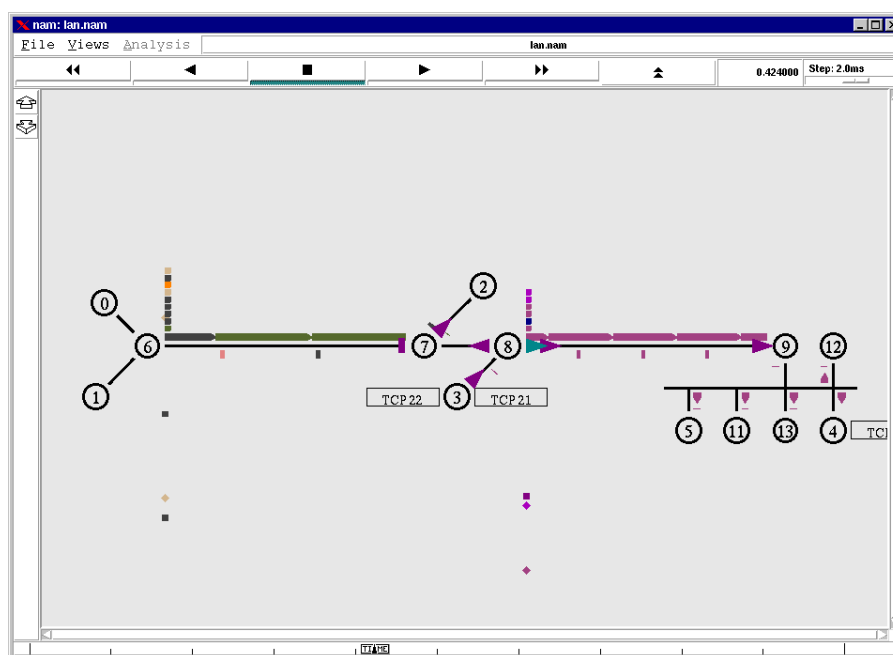
Slika 2.5: Izvajanje simulacije v simulatorju SUMO.

## 2.4 Izbira omrežnega simulatorja

Za simulacijo omrežja vozil so na voljo številni simulatorji. Sama simulacija zahteva dve komponenti: omrežno in mobilnostno. V večini primerov sta komponenti ločeni. Simulator toka prometa ustvari realni mobilnostni model, ki se nato uporabi v omrežnem simulatorju. Omrežni simulator izračuna in ustvari potrebne komponente v brezžičnem omrežju, kot je na primer podrobna struktura vseh vozil, pošiljanje in prejemanje paketov v omrežju. Omrežne simulatorje se običajno uporablja za simulacijo računalniških omrežij. Pri simulaciji omrežja vozil so zelo uporabni pri ocenjevanju zmogljivosti omrežnih protokolov za mobilnost vozlišč. Primer takšnih simulatorjev sta zelo priljubljena in razširjena simulatorja NS-2 in OMNeT++.

### 2.4.1 NS-2

NS-2 (angl. *Network Simulator 2*) je odprtokodni diskretni simulator. Uporablja se za simulacijo omrežnih protokolov za različne omrežne topologije. Možno je simulirati tako žična kot tudi brezžična omrežja. Napisan je v programskem jeziku C++ in omogoča simulacije preko vmesnika OTcl (angl. *Object-oriented Tool Command Language*, OTcl). Topologijo omrežja se sestavi preko skript OTcl, ki se nato z določenimi parametri izvedejo v NS-2. Za grafični prikaz omrežja se uporablja animator telekomunikacijskih omrežij (angl. *Network Animator*, NAM), ki je del simulatorja NS-2.



Slika 2.6: Izvajanje simulacije v NS-2 [18].



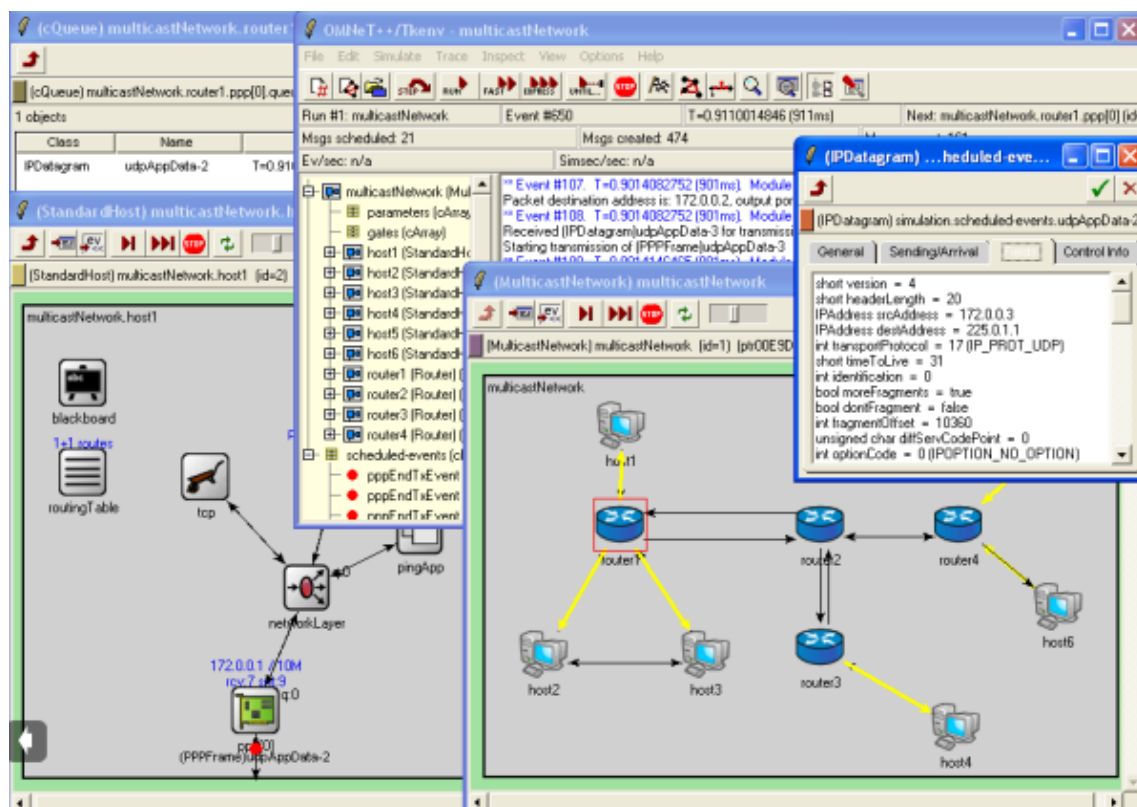
### 2.4.2 OMNeT++

OMNeT++ je odprtokodni diskretni simulator, napisan v programskem jeziku C++. Za razliko od simulatorja NS-2 ni namenjen samo simulacijam omrežij, ampak se ga uporablja tudi za modeliranje in oceno zmogljivosti distribuiranih sistemov. Najbolj pogosto področje uporabe je simulacija računalniških omrežij.

Simulacije je možno izvajati v okviru različnih uporabniških vmesnikov. Grafični, animirani uporabniški vmesnik je zelo uporaben za demonstracije in iskanje napak, za izvajanje skript pa se uporablja ukazna vrstica. Simulator omogoča tudi vzporedno, porazdeljeno izvajanje simulacij. Za izdelavo strukture simulacijskega modela se uporablja programski jezik NED (angl. *Network Description*). Ta omogoča uporabniku deklaracijo delov, t.i. modulov, ki se jih lahko poveže in sestavi v samostojni simulacijski model. Sporočila, kjer je podatkovna struktura poljubna, se med moduli prenašajo vzdolž vnaprej določene poti. Možna je tudi definicija parametrov, ki se lahko uporabijo za prilagoditev obnašanja modula tekom simulacije.

OMNeT++ zagotavlja osnovna orodja, ki sama po sebi ne prinašajo komponent za simulacijo računalniških omrežij. Tako so bili razviti številni modeli in ogrodja, kot so na primer INET, MiXiM in Veins.

Za izvajanje omrežnih simulacij smo izbrali omrežni simulator OMNeT++. V primerjavi z ostalimi simulatorji ima zelo dobro razvit grafični vmesnik, kompatibilen je z orodji, ki omogočajo analizo podatkov, ima razvite številne module, ki omogočajo komunikacijo z ostalimi vozili in za nas zelo pomembno dejstvo, da omogoča enostavno načrtovanje in spreminjanje simulacijskih scenarijev. Skozi delo nam je bil cilj ugotoviti čim bolj optimalen scenarij, kako medsebojno vplivata gostota vozil, opremljenih z napravami OBU, in pogostost pošiljanja pozdravnih sporočil. Pri tem nam je OMNeT++ zelo olajšal delo, ker sta sama nastavljivost števila vozil in pogostost pošiljanja pozdravnih sporočil v simulaciji zelo preprosti. Prav tako ga je enostavno povezati s cestnim simulatorjem omrežja SUMO. OMNeT++ omogoča vzporedno izvajanje simulacij s cestnim simulatorjem SUMO v realnem času.



Slika 2.7: Izvajanje simulacije v simulatorju OMNeT++ [19].

#### 2.4.2.1 INET

INET je ogrodje za OMNeT++, ki prinaša komponente in protokole, kot so TCP, UDP, IPv4, IPv6, Ethernet, IEEE 802.11 itd., za simulacijo žičnih, brezžičnih in mobilnih omrežij. Uporabno je predvsem pri oblikovanju in validaciji novih protokolov. Številna ostala simulacijska ogrodja vzamejo INET kot osnovo, ki se jo nato razširi v določene smeri, kot so omrežja vozil, omrežja točka-točka ali LTE.

INET je zgrajen kot koncept modulov, ki komunicirajo s pošiljanjem sporočil. Omrežni protokoli so predstavljeni kot sestavni deli, katere se lahko poljubno združuje in spreminja. Tako lahko tvorimo gostitelje, usmerjevalnike, stikala in druge omrežne naprave.

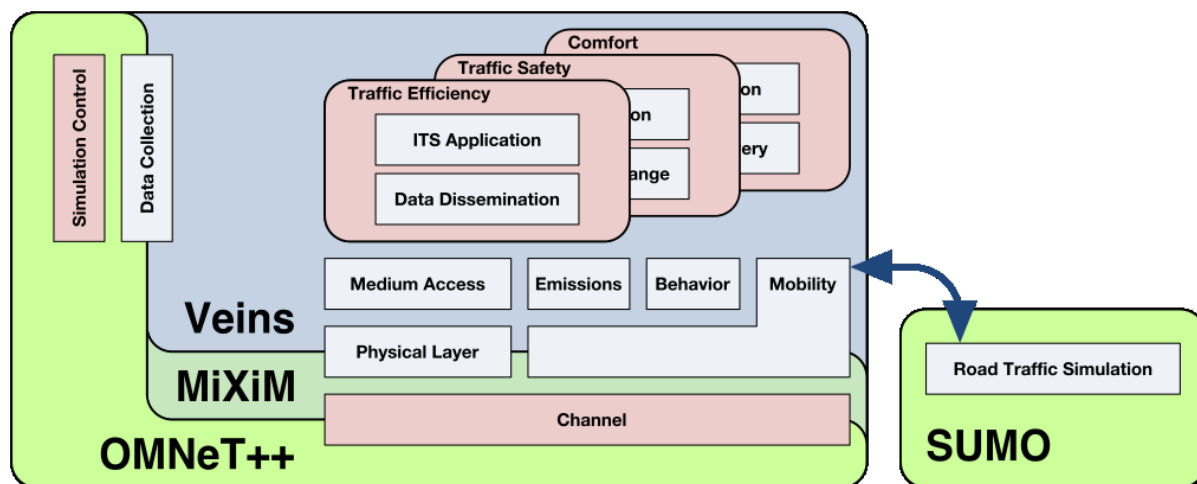
### 2.4.2.2 MiXiM

MiXiM je, podobno kot INET, ogrodje za OMNeT++. Ponuja podrobne modele, kot so širjenje radijskih valov, električna poraba energije radijskega sprejemnika in brezžični protokol MAC. Združen je iz več ogrodij za podporo simulacijam žičnih in brezžičnih omrežij.

### 2.4.2.3 Veins

Ogrodje Veins prinaša celovit nabor specifičnih modelov, ki omogočajo zelo realistične simulacije omrežij vozil.

Sestavljata ga dva različna simulatorja. Za simulacijo cestnega prometa se uporablja uveljavljen simulator SUMO. Simulacija omrežja se izvede z omrežnim simulatorjem OMNeT++, ki omogoča natančno simulacijo fizične plasti z ogrodjem MiXiM, vključno z radijskimi motnjami. Za uspešno komunikacijo med vozili morata oba simulatorja teči hkrati, povezana pa sta preko protokola za nadzor prenosa (angl. *Transmission Control Protocol*, TCP) in kontrolnega vmesnika prometa (angl. *Traffic Control Interface*, TraCI). Slika 2.8 prikazuje arhitekturo Veins.



Slika 2.8: Arhitektura Veins [22].

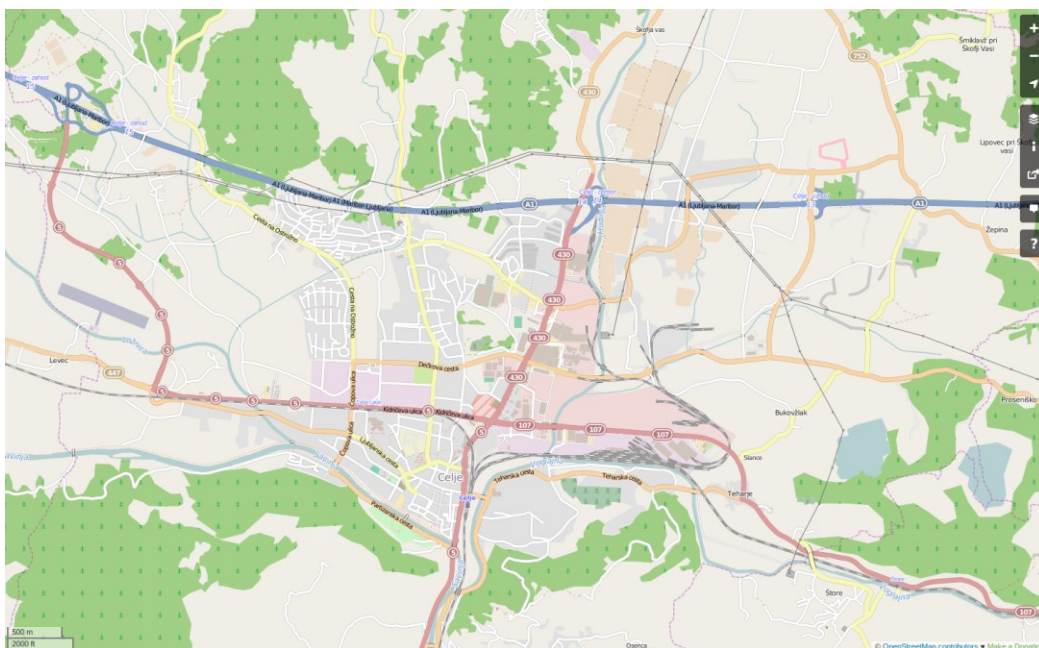


## **Poglavje 3 Priprava simulacijskega okolja**

Poglavje 3 opisuje pripravo simulacijskega okolja in konfiguracije obeh uporabljenih simulatorjev, ki jih potrebujemo za uspešno izvajanje simulacij. Cestno omrežje je opredeljeno v omrežni datoteki in tvori usmerjen graf. Ceste so opredeljene kot povezave v usmerjenem grafu in se jim lahko doda določene attribute, kot so omejitve hitrosti ali prepoved obračanja. Za generiranje poti oziroma kompleksnih omrežij lahko uporabimo vgrajene aplikacije. Najlažji način je generiranje naključnih poti. To pomeni, da lahko določimo različne izvirne lokacije vozil ob začetku simulacije, naključne zavoje na posameznih križiščih, različne hitrosti in čas gibanja vozil na cesti in različne končne cilje posameznih vozil. Količino prometa je možno nadzorovati z dodajanjem ali zmanjševanjem poti v določenem časovnem obdobju. Tako se lahko približamo realnemu cestnemu prometu.

### 3.1 OpenStreetMap

OpenStreetMap je priljubljena spletna storitev. Predstavlja brezplačno alternativo drugim kartografskim sistemom, ki jih najdemo na spletu. Zemljevidi so točni in se redno posodablajo, za kar skrbi skupnost projekta OpenStreetMap. Slika 3.1 prikazuje zemljevid Celja, katerega smo uporabili v naši simulaciji.



Slika 3.1: Zemljevid Celja.

## 3.2 Konfiguracija generatorja prometa SUMO

Zemljevid smo uvozili z OpenStreetMap. S pomočjo namenske aplikacije *NETCONVERT* smo pretvorili uvožen zemljevid v datoteko *celje.net.xml* in ga tako prilagodili za simulacijo v simulatorju SUMO. Slika 3.2 prikazuje pretvorjen zemljevid in cestno omrežje, ki ga bomo uporabili v simulaciji.



Slika 3.2: Pretvorjen zemljevid.

Da bi se približali realnemu scenariju omrežja vozil, smo morali pripraviti še model vozil in poti, po katerih se bodo vozila gibala. Vse te lastnosti pripravimo v specifičnih datotekah *xml*. Pri načrtovanju cestnega omrežja smo najprej pripravili datoteko naključnih poti za izbrano omrežje *celje.trips.xml*, katero prikazuje Koda 1. Datoteko smo ustvarili s pomočjo skripte *randomTrips.py*, ki jo dobimo pri namestitvi simulatorja SUMO. Poti dobijo unikatni id ter x in y koordinati.

---

**Koda 1:** Naključne poti v izbranem omrežju – del datoteke *celje.trips.xml*.

---

```
<?xml version="1.0"?>
<trips>
  <trip id="0" depart="0.00" from="-318965668" to="318965682#0" />
  <trip id="1" depart="1.00" from="172905697#4" to="304029923" />
  <trip id="2" depart="2.00" from="-244161729#1" to="8113133#5" />
  <trip id="3" depart="3.00" from="9236837" to="26305953#2" />
  <trip id="4" depart="4.00" from="170868814#0" to="9223103#6" />
  <trip id="5" depart="5.00" from="-225109263#1" to="178415258#1" />
  <trip id="6" depart="6.00" from="-278930012" to="314433620#5" />
  <trip id="7" depart="7.00" from="241817602#0" to="209434490#1" />
  <trip id="8" depart="8.00" from="-283533695" to="-244161724" />
  <trip id="9" depart="9.00" from="-194273355" to="241177202#3" />
  <trip id="10" depart="10.00" from="-243143771#1" to="-9223103#0" />
  .
  .
  .
</trips>
```

---



V naslednjem koraku smo ustvarili poti, po katerih se bodo vozila gibala. Uporabili smo aplikacijo *DUAROUTER*. Vsebino datoteke *celje.rou.xml* prikazuje Koda 2. Vozila dobijo unikaten id in poti, po katerih bodo vozila.

---

**Koda 2:** Gibanje vozil v omrežju – del datoteke *celje.rou.xml*.

---

```
<?xml version="1.0" encoding="UTF-8"?>

<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo-
sim.org/xsd/routes_file.xsd">

    <vehicle id="0" depart="0.00">
        <route edges="-318965668 -318965667#2 -318965667#1 -318965667#0
-13855465#4 -13855465#3 318965674 318965676#0 318965675 33792368#6 -
318965682#1 -318965682#0 318965682#0"/>
    </vehicle>

    <vehicle id="1" depart="1.00">
        <route edges="172905697#4 172905697#0 172905697#1 13202455#0
13202455#1 115506954 115506963#0 115506963#1 115506963#2 -207794067#4
33791792#0 -303051510#1 -303051510#0 -303051509#2 303052053#0
303052053#1 303052053#2 303052044#1 304029923"/>
    </vehicle>

    .
    .
    .
</routes>
```

---

Nato smo pripravili še konfiguracijsko datoteko za simulator SUMO (Koda 3), kjer smo poleg uvoženega zemljevida in poti določili tudi čas trajanja simulacije.

---

**Koda 3:** Konfiguracijska datoteka za simulator Sumo - *celje.sumo.cfg*.

---

```
<?xml version="1.0" encoding="iso-8859-1"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd"
">
    <input>
        <net-file value="celje.net.xml"/>
        <route-files value="celje.rou.xml"/>
    </input>
    <time>
        <begin value="0"/>
        <end value="900"/>
        <step-length value="0.1"/>
    </time>
    <gui_only>
        <start value="true"/>
    </gui_only>

</configuration>
```

---

### 3.3 Konfiguracija omrežnega simulatorja OMNeT++

Ogrodje Veins povezuje simulatorja SUMO in OMNeT++. V poglavju 3.2 smo pokazali, kako ustvariti realni promet in uvoziti digitalni zemljevid v SUMO. Z ogrođjem Veins smo implementirali protokol 802.11p v OMNeT++. Definicija omrežja je bila uvožena iz primera Veins iz ogrođja Veins. Sestoji se iz podmodulov:

- *ObstacleControl* za simulacijo blokade prenosa zaradi ovir,
- *AnnotationManager* za pridobivanje rezultatov/komentarjev,
- *ConnectionManager* za komunikacijo med vozlišči in
- *TraCIScenarioManagerLaunchd*, ki sproži in konča simulacijo na strežniku SUMO.

Sledila je priprava konfiguracijske datoteke *celje.launchd.xml* za OMNeT++, s katero smo definirali naše cestno omrežje.

---

**Koda 4:** Konfiguracijska datoteka za OMNeT++ – *celje.launchd.xml*.

---

```
<?xml version="1.0"?>

<launch>
  <copy file="celje.net.xml" />
  <copy file="celje.rou.xml" />
  <copy file="celje.sumo.cfg" type="config" />
</launch>
```

---

Za inicializacijo simulacije smo uporabili datoteko *omnetpp.ini*, kjer smo določili vrednosti parametrov in kako naj se simulacija izvede. Kot rezultate simulacij smo dobili tekstovne datoteke, katere smo kasneje uporabili za analizo.

V datoteki *omnetpp.ini* smo morali definirati območje, ki je bilo dovolj veliko in je vključevalo vse točke, katere smo uporabili v cestnem omrežju. Prav tako smo določili čas trajanja simulacije. Parametri simulacije so zbrani v kodi 5.

---

**Koda 5:** Parametri simulacije v datoteki *omnetpp.ini*.

---

```
debug-on-errors = true
print-undisposed = false

sim-time-limit = 900s

**.scalar-recording = true
**.vector-recording = true

**.debug = false
**.coreDebug = false

*.playgroundSizeX = 30000m
*.playgroundSizeY = 30000m
*.playgroundSizeZ = 50m
```

---

Pomembne so tudi nastavitve omrežne kartice in nastavitve protokola 802.11p katerega se uporablja za komunikacijo med vozili. Nastavitve, kot so oddajna moč, občutljivost sprejemnika, bitna hitrost in frekvenčni spekter, prav tako definiramo v datoteki *omnetpp.ini*.

---

**Koda 6:** Nastavitve omrežne kartice in protokola 802.11p v datoteki *omnetpp.ini*.

---

```
*.connectionManager.pMax = 20mW
*.connectionManager.sat = -89dBm
*.connectionManager.alpha = 2.0
*.connectionManager.carrierFrequency = 5.890e9 Hz
*.connectionManager.sendDirect = true

*.*.nic.mac1609_4.useServiceChannel = false

*.*.nic.mac1609_4.txPower = 20mW
*.*.nic.mac1609_4.bitrate = 18Mbps

*.*.nic.phy80211p.sensitivity = -89dBm
*.*.nic.phy80211p.maxTXPower = 10mW
*.*.nic.phy80211p.useThermalNoise = true
*.*.nic.phy80211p.thermalNoise = -110dBm
*.*.nic.phy80211p.usePropagationDelay = true
```

---



## Poglavje 4 Izvedba simulacije

V tem poglavju bomo predstavili izvedbo simulacij. Osnovna ideja je v povezavi TCP med zunanjo aplikacijo in simulatorjem SUMO, ki deluje kot strežnik, zunanja aplikacija pa kot odjemalec. V našem primeru je zunanja aplikacija omrežni simulator OMNeT++, ki se preko kontrolnega vmesnika prometa TraCI poveže s simulatorjem SUMO. Orodje, ki povezuje oba simulatorja, je ogrodje Veins. Tu se nahaja skripta *sumo-launchd.py*, s pomočjo katere simulator SUMO zažene strežnik TCP, ki posluša na portu 9999 in čaka na odjemalca TCP. Tako lahko ogrodje Veins vzpostavi sejo TCP in se poveže s simulatorjem SUMO. Ogrodje Veins izbere naključni izvorni port in uporabi port 9999 kot ciljni port. Nastavitve vmesnika TraCI definiramo v datoteki *omnetpp.ini*, kot je to prikazano v kodi 7.

---

### Koda 7: Nastavitve vmesnika TraCI.

---

```
# vmesnik TraCI

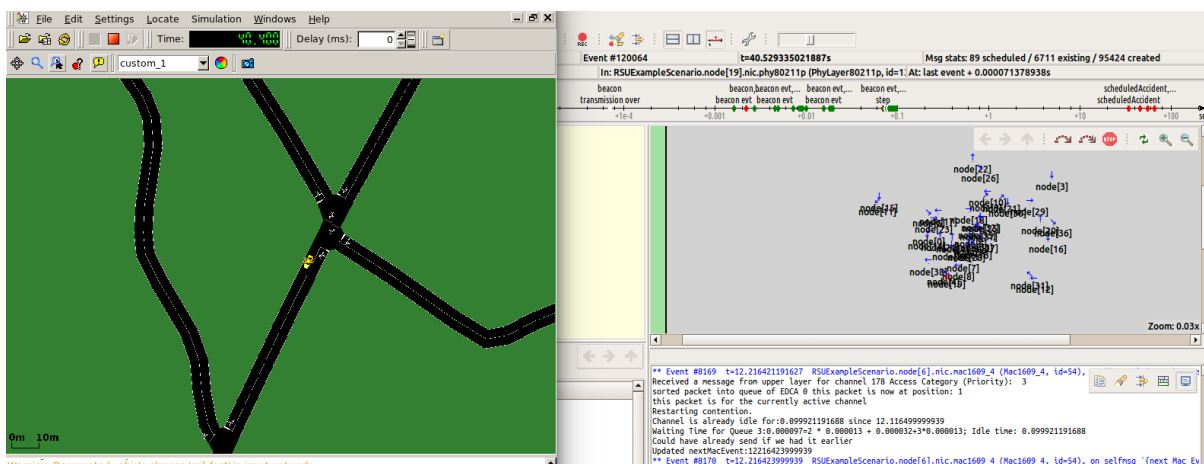
*.manager.updateInterval = 0.1s
*.manager.host = "localhost"
*.manager.port = 9999
*.manager.moduleType = "org.car2x.veins.nodes.Car"
*.manager.moduleName = "node"
*.manager.moduleDisplayString = ""
*.manager.autoShutdown = true
*.manager.margin = 25
*.manager.launchConfig = xmldoc("celje.launchd.xml")
```

---

Simulacije smo izvajali za omrežja vozil, ki so bila sestavljena iz 50, 100, 200 in 400 vozil, ki so se gibala po nastavitvah mobilnostnega modela, čas izvajanja simulacije pa je bil nastavljen na 900 sekund.

Vozila so bila opremljena s sprejemni enotami OBU, kar jim je omogočalo komunikacijo v brezžičnem omrežju preko protokola 802.11b.

Za vsa zgoraj omenjena omrežja vozil smo izvedli po pet simulacij, in sicer tako, da je vsako vozilo poslalo 2, 4, 6, 8 in 10 pozdravnih sporočil (angl. *beacon*) na sekundo, vsa sosednja vozila pa so lahko sprejela pozdravna sporočila.



Slika 4.1: Vzpostavljeno izvajanje simulacije v simulatorju SUMO in OMNeT++.



## **Poglavje 5   Rezultati simulacij**

Tekom izvajanja simulacije smo opazili, da je graf omrežja sestavljen iz več komponent. To pomeni, da v vsakem trenutku vsa vozila niso povezljiva. Razlog lahko pripišemo nastavitvam vozil. S tem ko smo se želeli čim bolj približati realnemu modelu omrežja vozil, smo vozilom določili različne začetne in končne položaje, hitrosti in čas vožnje. Večjo povezljivost bi lahko dosegli s povečanim številom vozil, z daljšim dometom sporočil ter s povečanjem oddajne moči OBU naprave.

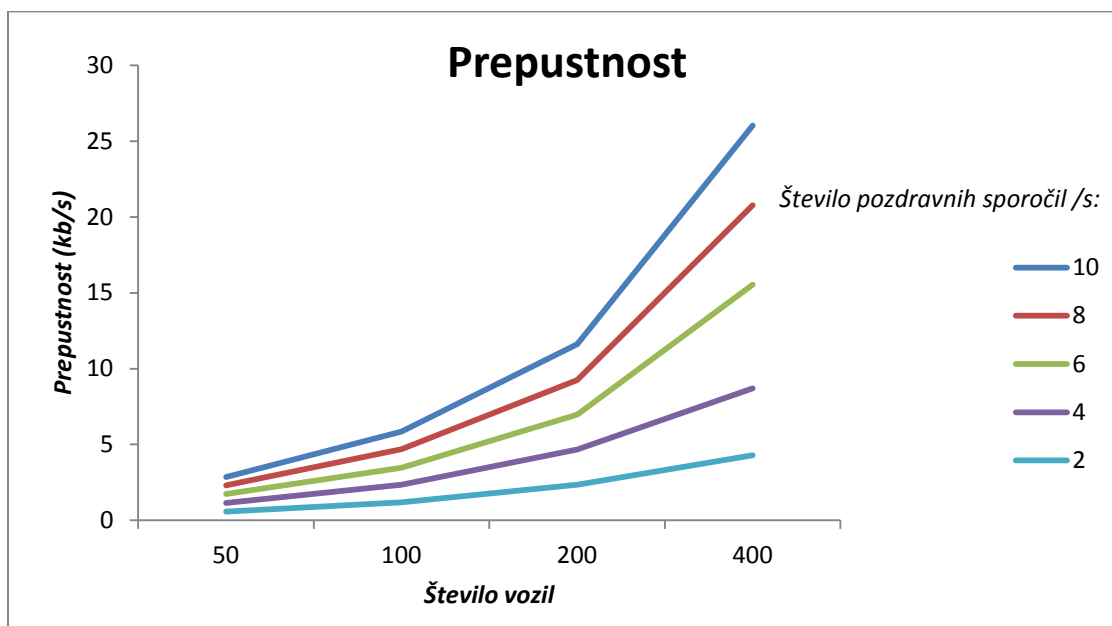
Za oceno zmogljivosti omrežja se uporablja različne parametre. Za oceno uporabljenega simuliranega omrežja smo uporabili prepustnost, izgubljene pakete in zakasnitev od konca do konca.

## 5.1 Prepustnost

Prepustnost (angl. *throughput*) je količina podatkov, ki se prenese skozi omrežje v enoti časa. Predstavlja število uspešno dostavljenih paketov v komunikacijskem omrežju. Na prepustnost vplivajo pogoste spremembe topologije omrežja, nezanesljiva komunikacija, omejena pasovna širina. Zaželena je visoka prepustnost v omrežju. Izračunana je bila po enačbi:

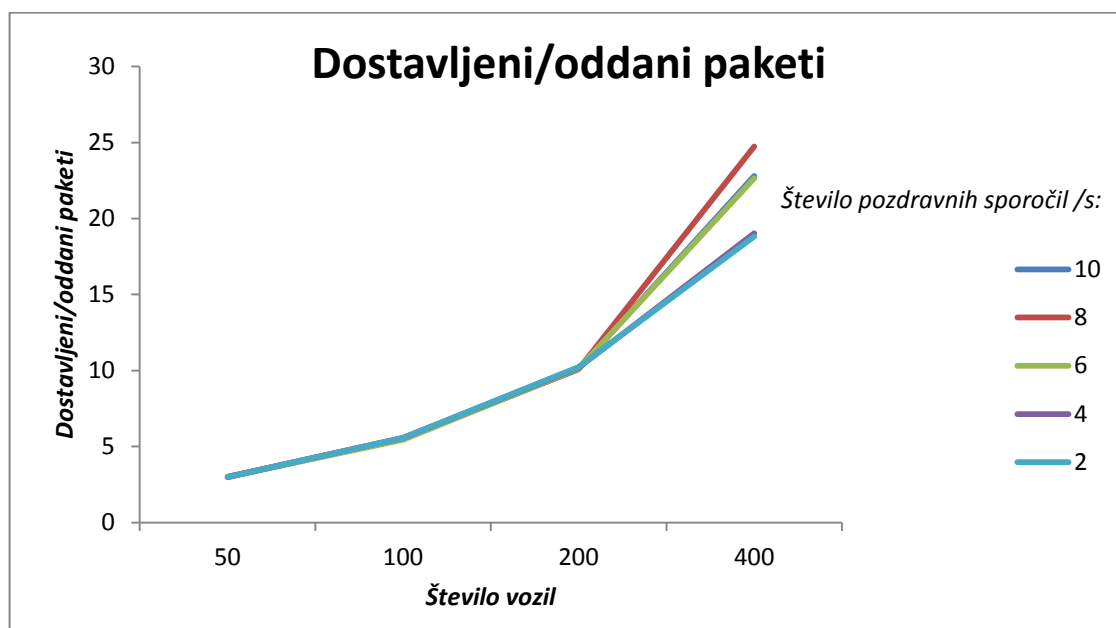
$$T = \frac{\text{število dostavljenih paketov} * \text{velikost paketa}}{\text{skupni simulacijski čas} * \text{število vozil}} \quad (5.1)$$

Iz slike 5.1 je razvidno, da se je prepustnost povečala z večjim številom vozil, največ podatkov pa se je preneslo skozi omrežje, ko so vozila pošiljala 10 pozdravnih sporočil na sekundo. K večji prepustnosti je pripomogla večja gostota vozil v simulacijskem območju, kar je vodilo do boljše povezljivosti med vozili.



Slika 5.1: Prepustnost.

Pogledali smo si tudi razmerje med številom dostavljenih in številom oddanih paketov pri vseh kombinacijah omrežij vozil. Dobljeni rezultati so predstavljeni na sliki 5.2.

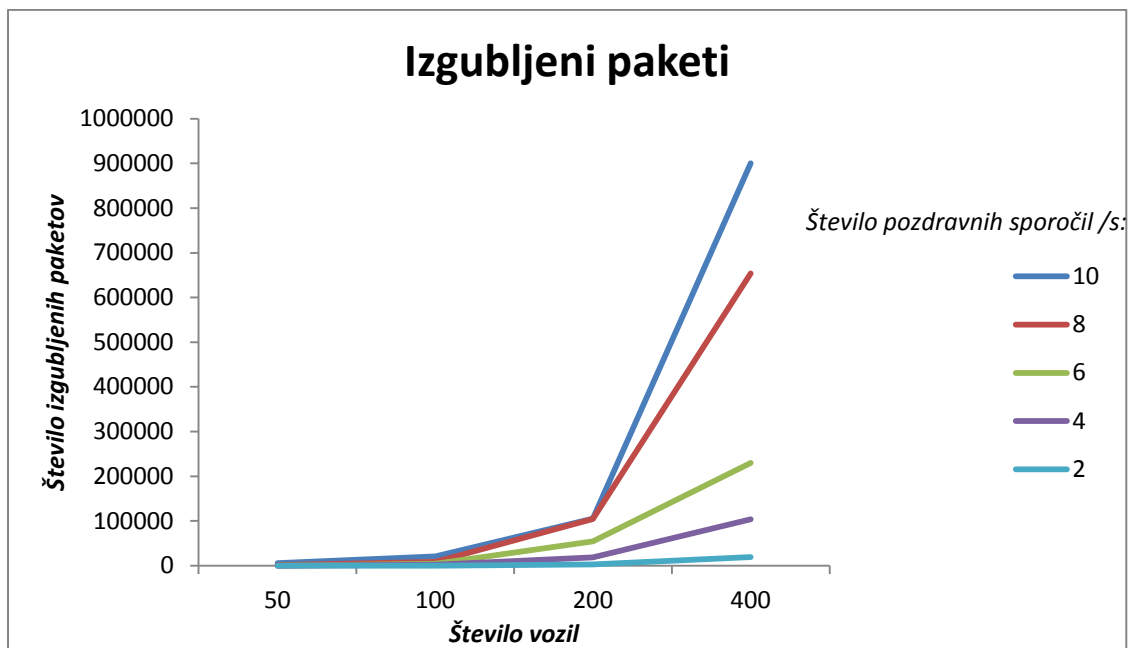


Slika 5.2: Razmerje dostavljenih in oddanih paketov.

S slike 5.2 je moč razbrati, da se število dostavljenih paketov glede na oddane povečuje s številom vozil. Največ dostavljenih paketov glede na oddane se prenese, če vozila pošiljajo 8 pozdravnih sporočil na sekundo.

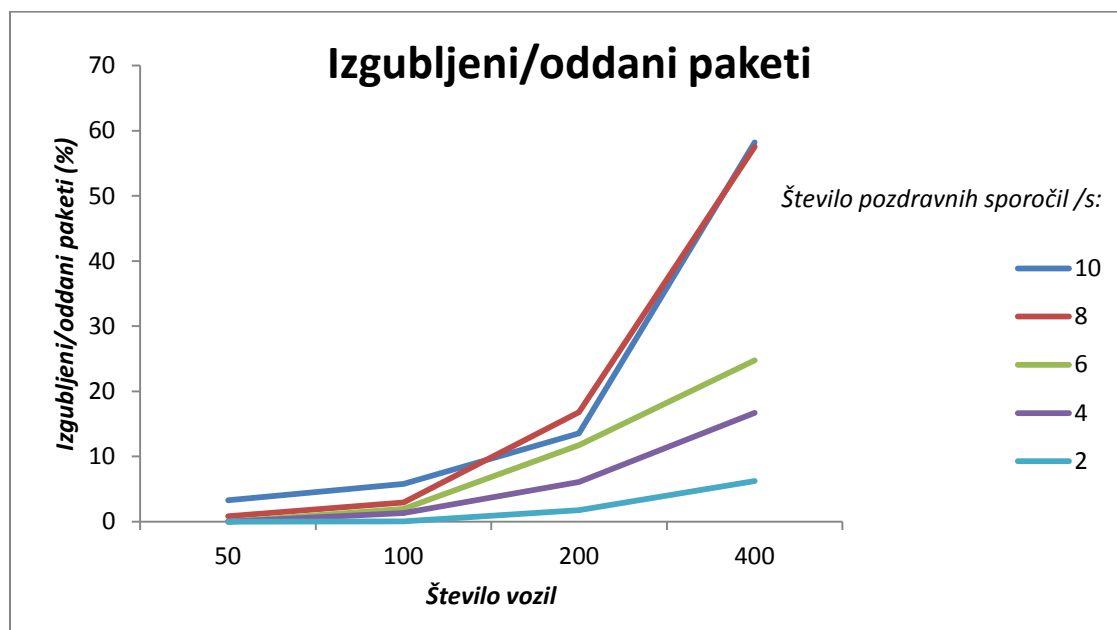
## 5.2 Izgubljeni paketi

Izgubljeni paketi so tisti paketi, ki niso dosegli cilja. Slika 5.3 prikazuje število izgubljenih paketov (angl. *packet drop*). Kot je razvidno s slike, se je število izgubljenih paketov močno povečalo z večjim številom vozil v omrežju. Paketi so se začeli izgubljati, ko so vsa vozila začela posredovati pozdravna sporočila, najmanj pa se jih je seveda izgubilo, ko so vozila pošiljala 2 pozdravni sporočili na sekundo. Paketi se lahko, na primer, izgubijo zaradi napake strojne opreme, zamašitve (angl. *congestion*) – stanje, ko ponujena obremenitev preseže zmogljivost sistema, prelivanja prometa (angl. *traffic overflow*) – stanje, ko ponujeni promet presega kapaciteto sistema in ga usmeri v alternativno smer.



Slika 5.3: Izgubljeni paketi.

Z grafa, ki ga prikazuje slika 5.4, vidimo, da se procent izgubljenih paketov glede na oddane povečuje s številom vozil, do največjih izgub paketov pride, ko vozila pošiljajo 8 ali 10 pozdravnih sporočil na sekundo.



Slika 5.4: Razmerje izgubljenih in oddanih paketov.

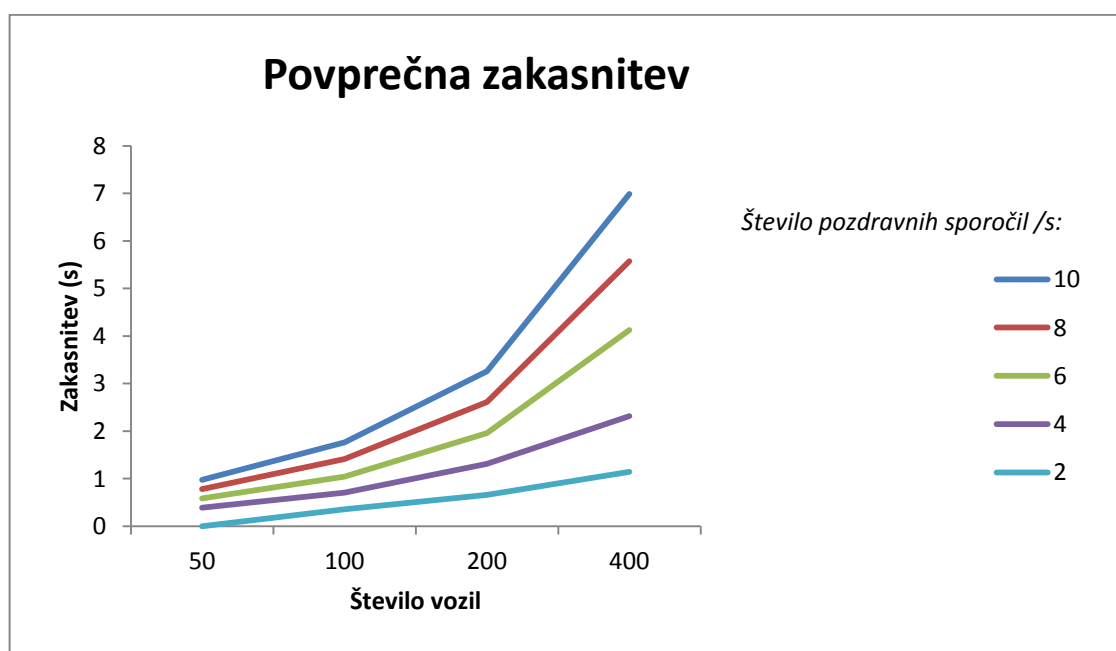
### 5.3 Zakasnitev od konca do konca

Meritev zakasnitve od konca do konca (angl. *End to End delay*) prikazuje zakasnitev v paketnem prenosu od izvirnega do končnega vozlišča. Vključuje različne zakasnitve, do katerih pride zaradi odkrivanja poti (angl. *route discovery*), čakalnih vrst, časa razširjanja (angl. *propagation time*) in prenosa paketov. Seveda si želimo čim manjše zakasnitve v omrežju. Za izračun smo uporabili enačbo:

$$\text{Povprečna zakasnitev od konca do konca} = \frac{T_R - T_S}{\text{število vozil}} \quad (5.2)$$

kjer je  $T_R$  čas, ko paketi prispejo do končnega vozlišča, in  $T_S$  čas poslanih paketov od izvirnega vozlišča.

Kot je razvidno s slike 5.5, se je povprečna zakasnitev z večjim številom vozil v omrežju povečala. Do tega pride zaradi pošiljanja pozdravnih sporočil sosedom in njihovega posredovanja pozdravnih sporočil nadaljnjim sosedom. Z večanjem števila vozil se vozilom poveča število sosedov in nadaljnjih sosedov, kar vodi do zastojev zaradi odkrivanja čedalje več novih poti in daljših čakalnih vrst. Najmanjše zakasnitve so bile pričakovano pri pošiljanju 2 pozdravnih sporočil na sekundo.



Slika 5.5: Povprečna zakasnitev.

V omrežjih vozil mora biti zakasnitev prenosa paketa zelo majhna. Informacije morajo čim prej priti od vozila do vozila, saj imajo tako vozniki dovolj časa, da reagirajo v dani situaciji. Ko se vozilo približuje križišču, mora zbrati informacije o ostalih vozilih, njihovi lokaciji in hitrosti. Tako lahko določi, ali se bodo njihove poti križale, in opozori voznike o morebitni nevarnosti.

Glede na dobljene rezultate simulacij lahko sklepamo, da bomo s prihodom tehnologije VANET v mestu Celje dosegli dovolj dobro povezljivost s štiristo vozili s pošiljanjem šestih pozdravnih sporočil na sekundo. S šestimi pozdravnimi sporočili na sekundo pri štiristo vozilih bi dosegli dovolj uspešno dostavljenih sporočil in razmeroma malo število izgubljenih paketov; sam prenos paketov pa bi potekal z dovolj majhnimi zakasnitvami. Z večjim številom vozil in poslanih pozdravnih sporočil bi prišlo do večjega zasičenja in s tem do večjih zakasnitev, kar bi vodilo v večje izgube dostavljenih paketov in slabše povezljivosti. Sčasoma, ko bo število vozil, opremljenih z napravami OBU, preseglo 400 vozil, bo potrebno pošiljati ustrezno manj pozdravnih sporočil na sekundo, da ne bo prihajalo do zasičenja. Zasičenju bi se lahko deloma izognili tudi s krajšim dometom pošiljanja sporočil.

## Poglavje 6 Zaključek

Razpoložljivost različne programske opreme, ki omogoča simulacijo, modeliranje in vrednotenje omrežij vozil in računalniških omrežij, je res velika. Med prebiranjem literature sem ugotovila, da imajo številni simulatorji določene pomanjkljivosti. Nekateri so omejeni samo na simulacijo določenega protokola, vozila ne morejo spreminjati smeri, nekateri nimajo grafičnega uporabniškega vmesnika, ki bi prikazal gibanje vozil, ali pa omrežni simulator in simulator prometa nista kompatibilna.

S pomočjo izbrane programske opreme smo predstavili delovanje in obnašanje protokolov ter obstoječih standardov v omrežjih vozil. V našem omrežju so bile poti, zavoji, hitrosti, čas gibanja vozil, izvirne in končne lokacije vozil naključno generirane, da bi se čim bolj približali realnemu cestnemu prometu. Seveda bi lahko pripravili tudi takšen testni scenarij, kjer bi bili ti parametri poljubno nastavljeni. Smiselno bi bilo tudi primerjati rezultate simulacij z rezultati pravih sprejemnih enot na vozilih, katerih pa na žalost nisem imela na voljo, prav tako pa bi bilo takšno testiranje s finančnega vidika zelo drago.

V simulaciji omrežja vozil je uporaba dobrega mobilnostnega modela kritičnega pomena saj so rezultati simulacije zelo odvisni od mobilnosti vozil. Z dobljenimi rezultati lahko ocenimo zmogljivost uporabljenega omrežja v simulaciji ter obnašanje pri povečanem številu vozil. Tako bi lahko na primer predvideli obnašanje prometa pri kakšnih večjih prireditvah. Rezultati tudi prikazujejo, da je verjetnost dostave pozdravnih sporočil različna pri različnem številu vozil. Z večanjem gostote vozil se je število izgubljenih paketov močno povečalo, prav tako pa se je povečal čas paketnega prenosa od izvirnega do končnega vozlišča. Kot najbolj optimalno glede na število dostavljenih in izgubljenih sporočil ter njihovih zakasnitev se je izkazalo pošiljanje šestih pozdravnih sporočil na sekundo, kjer smo dosegli dovolj uspešno dostavljenih in razmeroma malo izgubljenih sporočil. S pošiljanjem pozdravnih sporočil smo želeli doseči čim boljšo povezljivost vozil. Z večjo povezljivostjo vozil prejme več voznikov informacije o stanju v prometu in imajo tako dovolj časa, da odreagirajo v dani situaciji. S tem pa dosežemo tudi večjo varnost v prometu.

Razvoj tehnologije, protokolov, storitev v omrežjih vozil seveda teče dalje, saj je veliko problemov in izzivov še odprtih. Zelo aktiven pri razvoju je švedski proizvajalec vozil Volvo.



## Literatura

- [1] M. Amoozadeh, *Certificate Revocation List Distribution in Vehicular Communication Systems*, Stockholm, 2012. Dostopno na: <http://www.diva-portal.org/smash/get/diva2:612624/FULLTEXT01.pdf> (februar 2015).
- [2] ETH, *Vehicular Ad hoc Networks (VANET), Engineering and simulation of mobile ad hoc routing protocols for VANET on highways and in cities*, Zurich, 2004. Dostopno na: <http://e-collection.library.ethz.ch/eserv/eth:27335/eth-27335-01.pdf> (oktober 2014).
- [3] G. Ščavničar, S. Vodopivec, J. Bešter, *Usmerjevalni protokoli v brezžičnih omrežjih med vozili*, ERK 2013, Portorož, A:85-88. Dostopno na: <http://www.ltfe.org/wp-content/uploads/2013/09/scavnicarusmerjevalnip.pdf> (oktober 2014).
- [4] A. Štern, A. Kos, *Povezljivost v omrežjih VANET*, ERK 2012, Portorož, A:53-56. Dostopno na: [http://www.ltfe.org/wp-content/uploads/2012/10/sternpovezljivost\\_vp.pdf](http://www.ltfe.org/wp-content/uploads/2012/10/sternpovezljivost_vp.pdf) (oktober 2014).
- [5] LTFE, *ITKT Slovar*. Dostopno na: <http://slovar.ltfe.org/>.
- [6] ISO, *Standards*. Dostopno na: <http://www.iso.org/iso/home/standards.htm>.
- [7] Wikipedia, *IEEE 802.11p*. Dostopno na: [http://en.wikipedia.org/wiki/IEEE\\_802.11p](http://en.wikipedia.org/wiki/IEEE_802.11p) (februar 2015).
- [8] ITS Standards Program, *IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE)*. Dostopno na: <http://www.standards.its.dot.gov/factsheets/factsheet/80> (marec 2015).

- [9] A. Khan, S. M. Bilal, M. Othman, *A Performance Comparison of Network Simulators for Wireless Networks*, Kuala Lumpur. Dostopno na: <http://arxiv.org/ftp/arxiv/papers/1307/1307.4129.pdf> (januar 2015).
- [10] Venkatesh, A. Indra, R. Murali, *Routing Protocols for Vehicular Adhoc Networks (VANETs): A Review*, January 2014, ISSN 2079-8407. Dostopno na: [http://www.cisjournal.org/journalofcomputing/archive/vol5no1/vol5no1\\_4.pdf](http://www.cisjournal.org/journalofcomputing/archive/vol5no1/vol5no1_4.pdf) (marec 2015).
- [11] A. Singh, *Simulation and Analysis of AODV, DSDV, ZRP in VANETs*, Patiala, 2013. Dostopno na: <http://dSPACE.thapar.edu:8080/dSPACE/bitstream/10266/2213/1/Abhishek-801131030.pdf> (november 2014).
- [12] S. Rehman, M. A. Khan, T. A. Zia, L. Zheng, *Vehicular Ad-Hoc Networks (VANETs) - An Overview and Challenges*, Australia 2013, e-ISSN 2167-7336. Dostopno na: <http://article.sapub.org/10.5923.j.jwnc.20130303.02.html> (marec 2015).
- [13] AODV. Dostopno na: <http://moment.cs.ucsb.edu/AODV/> (marec 2015).
- [14] Wikipedia, *Ad hoc On-Demand Distance Vector Routing*. Dostopno na: [http://en.wikipedia.org/wiki/Ad\\_hoc\\_On-Demand\\_Distance\\_Vector\\_Routing](http://en.wikipedia.org/wiki/Ad_hoc_On-Demand_Distance_Vector_Routing) (marec 2015).
- [15] Main Page. Dostopno na: [http://sumo.dlr.de/wiki/Main\\_Page](http://sumo.dlr.de/wiki/Main_Page) (november 2014).
- [16] DLR, Institute of Transportation Systems. Dostopno na: [http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931\\_read-41000/](http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/).
- [17] *The Network Simulator – ns-2*. Dostopno na: <http://www.isi.edu/nsnam/ns/>.
- [18] *Nam: Network Animator*. Dostopno na: <http://www.isi.edu/nsnam/nam/>.
- [19] OMNeT++, *Discrete Event Simulator*. Dostopno na: <http://omnetpp.org/>.
- [20] INET Framework. Dostopno na: <http://inet.omnetpp.org/>.
- [21] MiXiM, *MiXiM project*. Dostopno na: <http://mixim.sourceforge.net/>.
- [22] Veins, *The open source vehicular network simulation framework*. Dostopno na: <http://veins.car2x.org/documentation/modules/>.

- [23] OpenStreetMap. Dostopno na: <http://www.openstreetmap.org/about>.
- [24] H. Noori, *Realistic Urban Traffic Simulation as Vehicular Ad-Hoc Network (VANET) via Veins Framework*, Tampere, ISSN 2305-7254. Dostopno na: <https://fruct.org/publications/fruct12/files/Noo.pdf> (januar 2015).
- [25] A. Ceoceca, R. Doumerc, G. Kadas, S. K. Mohammed, *Network planning of road side units in the intelligent transport system domain*, Aalborg, 2012. Dostopno na: [http://vbn.aau.dk/ws/files/72634886/NDS7GR720FINAL\\_PROJECT.pdf](http://vbn.aau.dk/ws/files/72634886/NDS7GR720FINAL_PROJECT.pdf) (januar 2015).
- [26] F. Valsangkar, *Design and Analysis of AODV Routing Protocol in VANET Using NS-2*, 2014, ISSN: 2321-9653. Dostopno na: <http://www.ijraset.com/files/serve.php?FID=929> (november 2014).
- [27] R. Pankaja, M. Shona, M. Archana, *Analysis of Routing Protocols for VANET using Real City Map Scenarios*, ICCTAC, 0975 – 8887. Dostopno na: <http://research.ijcaonline.org/icctac2014/number1/icctac1408.pdf> (november 2014).
- [28] A. K. Mahli, A. K. Verma, *Simulation and Analysis of Dropped Packets for DSR Protocol in VANETs*, IJARCET, 2012, 2278 – 1323. Dostopno na: <http://ijarcet.org/wp-content/uploads/IJARCET-VOL-1-ISSUE-4-267-272.pdf> (november 2014).
- [29] S. Saha, U. Roy, D. D. Sinha, *AODV Routing Protocol Modification with Broadcasting RREQ Packet in VANET*, 2014, ISSN 2250-2459. Dostopno na: [http://www.ijetae.com/files/Volume4Issue8/IJETAE\\_0814\\_71.pdf](http://www.ijetae.com/files/Volume4Issue8/IJETAE_0814_71.pdf) (november 2014).
- [30] V. Singla, N. Wadhwa, *Comparative Performance Analysis of DSDV, AOMDV and ZRP Routing Protocols in VANET using NS2*, 2014, 2238-2241. Dostopno na: <http://www.ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502288.pdf> (november 2014).

- [31] M. J. Prokop, *Routing Protocol Evaluation and Development Of A Fully Functional Simulation Environment For Vehicular Ad Hoc Networks*, New York, 2011. Dostopno na:  
<https://ritdml.rit.edu/bitstream/handle/1850/14521/MProkopThesis8-2011.pdf?sequence=1> (januar 2015).